BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PATENT SPECIFICATION (11) 1 548 497

1 548 497

(21) Application No. 15763/77 (22) Filed 15 April 1977

(31) Convention Application No. 681 364 (32) Filed 29 April 1976 in

(33) United States of America (US)

(44) Complete Specification published 18 July 1979

(51) INT. CL.[3] G06F 9/06

(52) Index at acceptance G4A 17B 17P 5A 5B 9C 9X FL

(19)

(54) DATA PROCESSING SYSTEM

(71) We, NCR CORPORATION of Dayton in the State of Ohio, and Baltimore in the State of Maryland, United States of America, a corporation organized under the laws of the State of Maryland, United States of America, do hereby declare the invention, for which we pray that a patent may be granted to us, and the method by which it is to be performed, to be particularly described in and by the following statement:—

This invention relates to data processing systems.

One of the most substantial, and perhaps the dominant, cost in an operative data processing system is found in the software. As new generations of hardware have become available, it has become apparent that there is too much time and money invested in current software and user programs to simply disregard in favour of new-generation hardware. Moreover, a user faced with a massive effort to upgrade to a new system is vulnerable to the suggestion that he can just as easily switch to a new computer supplier with possible cost/performance gains if the already-developed software must be discarded. Thus, it will be apparent that a prime requirement for new generations of computer hardware is true compatability with already-developed software.

According to the present invention, there is provided a data processing system including a plurality of firmware-configurable modules, non-volatile auxiliary storage means storing firmware information for said modules, program storage means storing an access program for accessing information stored by said auxiliary storage means, and processing means, the arrangement being such that, in operation, in response to an initiating signal, said access program causes a loader routine to be rendered effective to cause said processing means to transfer said firmware information read from said auxiliary storage means to said firmware-configurable modules to thereby configure said modules.

By a firmware-configurable module herein is meant a unit adapted to perform operations of the kind utilized in data processing systems and including an alterable microprogram control section adapted to control the operations performed by the unit when firmware information has been entered into the microgram control section to configure the module.

It will be appreciated that a data processing system according to the preceding paragraph but one has the advantage of being readily adapted to emulate another data processing system by creating a "virtual" machine according to the particular firmware information with which the firmware-configurable modules are configured.

One embodiment of the invention will now be described by way of example with reference to the accompanying drawings, in which:—

Figure 1 is a block diagram of a data processing system; and

Figure 2 is a more detailed block diagram of a part of the data processing system shown in Figure 1.

Attention is now directed to Figure 1 which illustrates, in a block diagram, a data processing system. The processing system of Figure 1 is bus-oriented in that various subsystems, including the processor subsystem 1, are coupled to one another by means of at least one internal transfer bus 2. The internal transfer bus 2 is incorporated into an internal transfer bus subsystem 3 which also includes a plurality of local bus adapters 4, an inter-bus communication adapter 5, bus control logic 6, and timing logic 7. In the internal transfer bus subsystem, up to sixteen local bus adapters may be coupled to the internal transfer bus 2. Undesignated subsystems 8 may con-

stitute any of the typical subsystems usually found in a bus-oriented data processing system; e.g.: core memory, tape units, disc units, printers, video displays, etc. The service subsystem 9 may include a system console to effect two-way communication between an operator and the data processing system.

The function of the internal transfer bus 2 is to transmit information from one local bus adapter to another local bus adapter. All information is passed over the internal transfer bus sequentially following a common procedure. The identical local bus adapters 4 interface each subsystem to the internal transfer bus 2. Each local bus adapter performs all the logic operations necessary to ensure that an internal transfer bus discipline is maintained at all times.

The bus control logic 6 performs several functions. It arbitrates all local bus adapter requests to use the internal transfer bus on a fixed priority basis. The bus control logic also checks the parity of all messages sent over the internal transfer bus and signals the result of the parity check onto the internal transfer bus for analysis by the communicating local bus adapters. The bus control logic can communicate with the service subsystem 9 and the processor subsystem 1 by means of a serial service bus 10 in order to provide certain system condition history and status information and configuring signals. The internal transfer subsystems are arraged such that all operations in the internal transfer subsystems are synchronized with clock and phase signals emanating from the timing logic 7.

The inter-bus communications adapter 5 facilitates communications between the system components associated with the interal transfer bus subsystem 3 and the subsystems associated with a second internal transfer bus subsystem 3' to which additional subsystems 8' are connected.

Figure 2 includes a block diagram of the service subsystem 9. As instruction counter 12 is a sixteen-bit counter which addresses an instruction memory 14 whereby the resultant memory data is made available to the service processor. Two modes of operation are possible for the instruction counter 12; viz.; increment and parallel load. Normally, the instruction counter addresses consecutive memory locations in the instruction memory 14 by means of the increment mode. The parallel load mode is used to load an address into the instruction counter during a branch operation to a non-consecutive memory location and also to effect system initialization as will be exaplained more fully below.

The instruction memory 14 accepts a sixteen-bit memory address. and there are 65,535 available memory addresses. The first 1024 memory addreses are allocated to a non-volatile read-only memory section 16 of the instruction memory, and the remaining memory is contained in a read/write random access section 18. The number of random access memory words actually implemented will depend on the characteristics of the given system in which the service subsystem 9 is resident.

Instruction register 20 is an eighteen-bit register which receives the sixteen-bit instruction being executed as well as one parity bit on each of two instruction bytes. Instruction decoding logic 22 is employed to decode the operation code portion of the instruction temporarily stored in the instruction register 20. The instruction decoding logic 22 provides the control signals necessary to execute the specific instruction identified by the opcode. In a presently preferred embodiment of the service subsystem 9, programmable ready-only memory chips are employed to effect the decoding in the instruction decoding logic 22.

The service subsystem 9 contains sixteen general purpose registers contained in a register memory 24. These general purpose registers are allocated for use as accumulators, index registers, counters, etc. All arithmetic, shift, load and store operations involve one or more of these registers.

The basic logic structure of the service processor is oriented around a sixteen-bit arithmetic and logic unit 26. In the presently preferred embodiment of the service processor, the arithmetic and logic unit 26 includes a plurality of integrated circuit chips.

Data is transferred into and out of the service processor through an external register unit 28. The addressing structure employed allows a possible 128 external registers in the external register unit 28. Communication to the internal transfer bus subsystem 2 (Figure 1) is effected by a local bus adapter 30 which is contained within the service processor 9. Additionally, the registers of the external register unit 28 may be employed to communicate with a plurality of peripheral subsysetems represented by the logic block 32. During normal data processing system operation, the service subsystem 9 is utilized as an input/output controller disposed between the peripherals represented by the block 32 and the internal transfer bus 2 and the subsystems coupled thereto.

It will be observed that the service subsystem 9 can communicate with the system processor 1 through the external register unit 28, the local bus adapter 30, and the internal transfer bus 2. In addition, however, more direct communication between the service subsystem 9 and the system processor 1 may be carried out over the serial

service bus 10.

The external register unit 28 is also utilized to couple an operator's console 34 to the service processor. The primary switches and indicators for system start-up and control are located on the operator's console 34. The operator's console 34 is also connected to the instruction counter 12 in order to force the instruction counter 12 to a predetermined memory location in the read only memory section 16 of the instruction memory 14 upon system initialization as will be discussed below. An interrupt logic block 38 also has the capability for forcing the instruction counter 12 to predetermined locations in response to signals received external to the service processor 9.

Firmware for the entire system is stored on a movable magnetic medium such as a flexible disc (sometimes called a " diskette ") which, with its associated electronics, is represented by the block 36 in Figure 2. The flexible disc unit may be manually removed and replaced by another disc unit.

When power to the system has been interrupted, whether deliberately or unexpectedly, the firmware within the firmware logic 44 of the system processor 1, the random access memory portion 18 of the instruction memory 14 (within the service processor), and other volatile firmware storage means in the diverse modules 40 will have been destroyed. These firmware configured modules are thus completely inoperative at this time.

System startup is initiated by acuation of a power control switch (not shown) which is located on the console 34 or in any other convenient physical location. In addition to the initiation of the usual powering-up activity such as pulling in power relays, starting the motors of rotating memories, etc., actuation of the power control switch issues a signal to the instruction counter 12 within the service processor to force the instruction counter to a predetermined memory storage address (such as 00000) located within the read-only memory portion 16 of the instruction memory 14. The predetermined memory storage location contains the first instruction of a simple bootstrap routine which functions, after determining that the disc 36 has attained operating speed, to call in a loader routine from the disc 36. As is well known, a bootstrap routine consists of preliminary pre-set instructions which call in other instructions to read in programs and data. This loader routine passes through the external register unit 28 into the random access memory portion 18 of the instruction memory 14. When the entire loader routine has been obtained from the disc 36, the bootstrap routine branches to the first instruction of the loader routine which assumes control of the service processor and configures the firmware-configurable modules 40 with firmware information from the disc 36. The loader routine is tailored to the individual system which it is prepared to configure: i.e., the loader routine contains, or has access to, information as to what specific firmware-configurable modules are coupled to the internal transfer bus 2 at specific ports of the bus.

The firmware logic 44 can be configured either through the serial service bus 10 or the internal transfer bus 2 whereas all other firmware-configurable modules must be configured from the information contained on the disc 36 through the internal transfer bus 2.

When all firmware configurable modules in the system, including the system processor, have been configured with the firmware information from the disc 36, the service processor relinquishes system control to the system processor 1. At this time, the service processor itself is reconfigured to function as an input/output controller for the peripherals 32 by overlaying the loader routine in the random access memory portion 18 of the instruction memory 14 by the appropriate firmware from the disc 36. Normal operation is then instituted.

It will be apparent that dynamic reconfiguration can be achieved by, for example, generating an interrupt signal from any appropriate location in the data processing system to the interrupt logic 38 of the service processor. If that specific interrupt forces the instruction counter 12 to the predetermined memory storage location in the read-only memory 16 portion of the instruction memory 14 constituting the first instruction in the bootstrap routine, the system will be reconfigured in the manner previously described with the firmware information contained on the disc 36 which, it will be understood, can be readily changed by simply substituting one disc unit for another prior to the occurrence of an interrupt or reinitialization.

WHAT WE CLAIM IS:—

1. A data processing system including a plurality of firmware-configurable modules, non-volatile auxiliary storage means storing firmware informaion for said modules, programe storage means storing an access program for accessing information stored by said auxiliary storage means, and processing means, the arrangement being such that, in operation, in response to an initiating signal, said access program causes a loader routine to be rendered effective to cause said processing means to transfer said firmware information read from said auxiliary storage means to said firmware-configurable modules to thereby configure said modules.

2. A data processing system according to

Claim 1, wherein said program storage means includes a non-volatile read-only memory section storing a bootstrap routine, and a read/write memory section, said auxiliary storage means storing said loader routine for controlling firmware information transfer, whereby in operation, in response to said initiating signal, said bootstrap routine is rendered effective to cause said processing means to transfer said loader routine from said auxiliary storage means to said read/write memory section.

3. A data processing system according to Claim 2, wherein said processing means includes an instruction counter adapted to control the addressing of said program storage means said initiating signal being effective to cause said instruction counter to be set to an initial address of said bootstrap routine

4. A data processing system according to Claim 2, or Claim 3, wherein said loader routine when stored in said read/write memory section is effective to cause said processing means to transfer said firmware information read from said auxiliary storage means to said firmware-configurable modules to thereby configure said modules.

5. A data processing according to Claim 4, including on input/output controller forming one of said firmware-configurable modules and including said processing means and said program storage means, said loader routine being adapted to effect the overlaying of said loader routine by firmware information for said input/output controller.

6. A data processing according to any one of the preceding Claims, including a power control switch adapted when operated to provide said initiating signal.

7. A data processing system substantially as hereinbefore described with reference to the accompanying drawings.

D. MILLICHAP,
Chartered Patent Agent,
Agent for the Applicant.

Printed for Her Majesty's Stationery Office by The Tweeddale Press Ltd., Berwick-upon-Tweed, 1979.
Published at the Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

1 548 497 COMPLETE SPECIFICATION
2 SHEETS

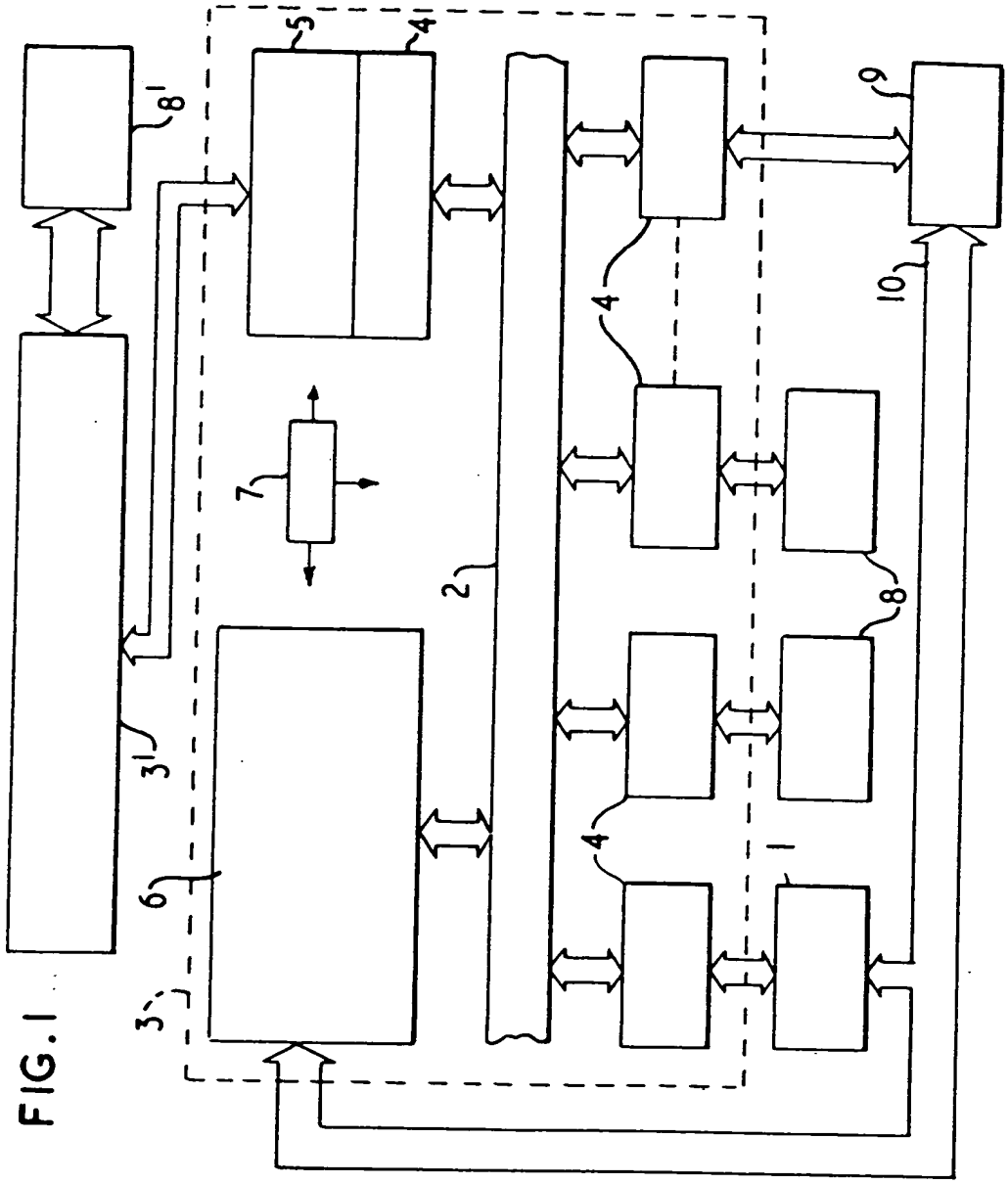

FIG. 1

1 548 497 COMPLETE SPECIFICATION
2 SHEETS This drawing is a reproduction of the Original on a reduced scale.
SHEET 2

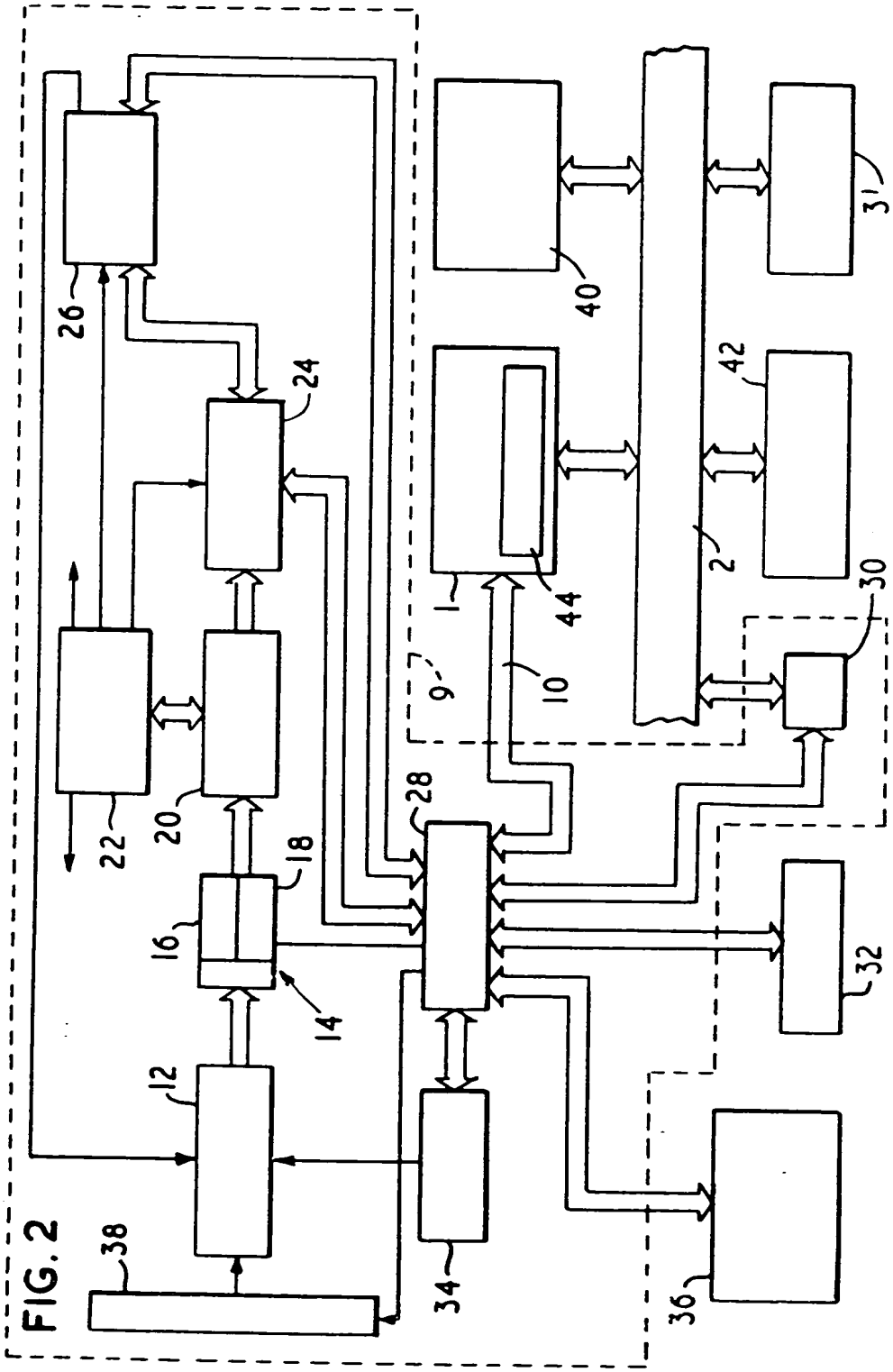

FIG. 2